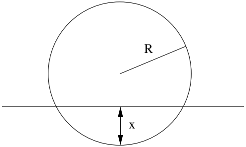
Ricerca di zeri di equazioni non lineari

Problema (esboa).
Si vuole determinare l'altezza x della parte sommersa di una boa sferica di raggio $R = 0.055m$ e densità di massa $\rho_b = 0.6Kg/m^3$, posta in acqua e soggetta alla sola forza peso.



Modello: terza legge di Newton

$$\begin{aligned} |\text{forza peso}| &= |\text{spinta idrostatica}| \\ m_b g &= m_{acquaspostata} g \\ V_b \rho_b g &= V_{calotta} \rho_w g \end{aligned}$$

$$(m = V\rho,\ g = 9.8\ \mathrm{m/s^2})$$

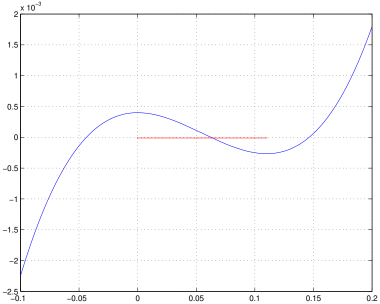
Indicando con $\rho_w = 1$ la densità dell'acqua (a condizioni opportune) e con ρ_b la densità della boa, si ha:

$$\frac{4}{3}\pi R^3 \rho_b g = \pi x^2 \left(R - \frac{x}{3}\right) \rho_w g. \tag{1}$$

Semplificando si ha l'equazione non lineare algebrica di terzo grado per x:

$$x^3 - 3x^2R + 4R^3\rho_b = 0.$$

La soluzione x del problema è la radice della funzione $f(x) = x^3 - 3x^2R + 4R^3\rho_b$ compresa nell'intervallo $(0, 2R)$.
La funzione $f(x) = x^3 - 3x^2R + 4R^3\rho_b$ ha il grafico seguente.



In rosso è stato evidenziato l'intervallo di accettabilità delle soluzioni.

Risolvere il problema numericamente

1. Definire la funzione di cui calcolare la radice e localizzare le radici per via grafica (scrivere un m-file).
2. Risolvere l'equazione non lineare con il metodo di bisezione (chi non ha seguito la lezione scarichi `bisezione.m` da `http://paola-gervasio.unibs.it/CS/matlab`)
`help bisezione`.
Fissare:
- intervallo iniziale $[a, b]$ =intervallo di accettabilità della soluzione,
- tolleranza per il test d'arresto pari a 10^{-8};
- numero massimo di iterazioni pari a 100.
3. Il numero di iterazioni effettuate dal metodo di bisezione concorda con quanto ci si aspetta dalla teoria?

Osservazioni

Dal comando plot (o fplot) si evince che:

- c'è una radice in prossimità di 0.06.
- ci sono altre due radici fuori dall'intervallo $(0, 0.11)$.

```
help bisezione
a=0; b=0.11; kmax=100; tol=1.e-8;
R=0.055; rho=0.6;
f=@(x)x.^3-3*x.^2*R+4*R^3*rho
% matlab e' in grado di sostituire a R e rho il loro
% valore all'interno di f
[z,res,it]=bisezione(f,a,b,tol,kmax)
```

Si ottiene:

```
z =
    0.0624
res =
   4.7070e-11
it =
    23
```

Risposta al punto 3.
Indicando con $c^{(k)}$ il punto medio dell'intervallo k-simo, ovvero l'approssimazione di α al passo k, la teoria dice che il numero di iterazioni di bisezione che garantisca di soddifare la stima $|c^{(k)} - \alpha| \leq \epsilon$ è:

$$k \geq \log_2\left(\frac{b-a}{\epsilon}\right) - 1$$

Noi otteniamo....

Metodo di Newton

$$\begin{cases} x^{(0)} \text{ dato} \\ x^{(k+1)} = x^{(k)} - \dfrac{f(x^{(k)})}{f'(x^{(k)})}, \quad k = 0, 1, \ldots. \end{cases}$$

Test d'arresto: $|x^{(k+1)} - x^{(k)}| < \varepsilon$

Algoritmo: Non serve memorizzare tutte le $x^{(k)}$, ma basta tenere in memoria solo due valori: $x^{(k)}$ (in `x`) e $x^{(k+1)}$ (in `xnew`)

```
err=1; k=0; x=x0; zv=[x0];
while k< kmax && err> tol
  valuta f, df e calcola xnew=....
  aggiorna err=|xnew-x| aggiorna k; aggiorna x
end
```

INPUT: `f, df, x0, tol, kmax`

OUTPUT: `z, res, it`

dove `z`=ultima $x^{(k+1)}$ calcolata, `res`=$f(x^{(k+1)})$, `it`=numero di iterazioni effettuate in tutto.

Metodo delle secanti

$$\begin{cases} x^{(0)}, x^{(1)} \text{ dati} \\ \\ x^{(k+1)} = x^{(k)} - \dfrac{f(x^{(k)})}{\frac{f(x^{(k)})-f(x^{(k-1)})}{x^{(k)}-x^{(k-1)}}}, \quad k = 1, 2, ... \end{cases}$$

Test d'arresto: $|x^{(k+1)} - x^{(k)}| < \varepsilon$

Implementazione: Modificare opportunamente la function `newton.m` (inclusi gli input).

Problema esboa. Seconda parte.

1. Risolvere il problema della boa con il metodo di Newton scegliendo opportunamente $x^{(0)}$, tolleranza 10^{-8}, numero massimo di iterazioni pari a 100. Calcolare $f'(x)$ a mano.
2. Lanciare Newton con tre dati iniziali diversi:

```
1.   x0=0;
2.   x0=0.03;
3.   x0=0.01;
```

Come si comporta il metodo? Converge? A cosa? Perchè? Dare un'interpretazione dei risultati.
3. Confrontare Newton e bisezione in termini di velocità di convergenza (numero di iterazioni) e di accuratezza della soluzione
4. La radice da calcolare è semplice o multipla? Cosa ci si aspetta dal teorema di convergenza di Newton? I risultati numerici riflettono quanto dice la teoria?

Conclusioni sul metodo di Newton

Nel primo caso la function si arresta subito perchè $f'(0) = 0$.
Nel secondo caso si giunge a convergenza in 5 iterazioni.
Nel terzo caso si arriva a convergenza in 6 iterazioni, ma non alla radice $x = 0.06237758...$, bensì a $x = 0.14635950..$, perchè ....

Problema esboa. Terza parte.

1. Risolvere il problema della boa con il metodo di secanti scegliendo opportunamente $x^{(0)}$, $x^{(1)}$, tolleranza 10^{-8}, numero massimo di iterazioni pari a 100.
2. Lanciare secanti con $x^{(0)}$, $x^{(1)}$ opportuni e confrontare i risultati ottenuti con quelli prodotti dal metodo di Newton.